

SAG sprawozdanie

Symulacja wykorzystania i przepustowości wyjść awaryjnych

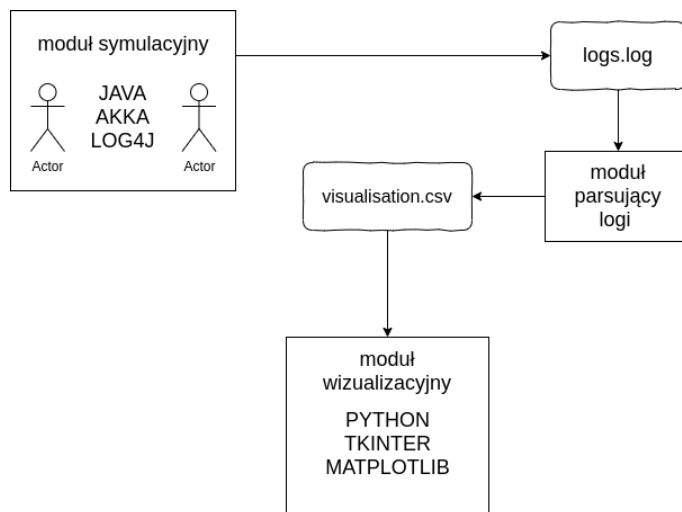
Czerwiec 2020

1 Autorzy

- Maria Skarbek
- Krzysztof Apolinarski
- Miłosz Parszewski

2 Wstep

Celem projektu było przygotowanie programu symulującego ewakuację z budynku w którym zaczął się pożar. Przy realizacji przyjęte zostało, że w całym budynku uruchamiają się alarmy więc nawet osoby niewidzące ognia będą uciekać.



Rysunek 1: Architektura systemu

Od strony technicznej sam projekt został podzielony na dwie podstawowe części co widać na powyższym diagramie. Pierwszym modulem jest “moduł symulacyjny” napisany z wykorzystaniem zestawu narzędziowego Akka w języku Java. Moduł ten przeprowadza symulacje na podstawie pliku w formacie .yaml przechowującego strukturę budynku. Wyniki symulacji umieszczane są w trakcie wykonywania przez agentów w pliku logs.log ,który to po zakończeniu całej symulacji zamieniany jest do formatu .csv.

Po zakończeniu wykonania symulacji przez “moduł symulacyjny” uruchamiany jest “moduł wizualizacyjny” ,który napisany został w języku Python z wykorzystaniem przede wszystkim bibliotek takich jak tkinter i matplotlib. Moduł ten wizualizuje przebieg ewakuacji na podstawie danych z pliku .csv

3 Przebieg Symulacji - Moduł Symulacyjny

W module symulacyjnym tworzony jest najpierw agent odpowiedzialny za zarządzanie całym procesem symulacji. Nazywa się on “Controller”. Następnie główny plik wykonawczy Main.java wykonuje pierwszy dostęp do klasy Building.java czym inicjalizuje ten singleton. Proces ten uruchamia również odczytanie informacji z pliku yaml. W kolejnym kroku tworzona jest grupa agentów poprzez wysłanie do Controllera wiadomości “CreateActor”. W tym procesie pracuje HumanConfigGenerator odpowiedzialny za nadanie aktorom statystyk - część z nich jest losowana.

W tym momencie system jest gotowy do symulacji co się odbywa poprzez pętlę wykonującą się tyle razy ile rund zostało przewidzianych. Każda runda składa się z rozprzestrzenienia się ognia ,który ma określoną w programie (aktualnie 20%) szansę na rozprzestrzenienie się w losowym kierunku z dowolnego pola ognia. W każdej rundzie wszyscy aktorzy wykonują swoją turę.

Po zakończeniu wszystkich iteracji pętli uruchamiany jest fragment kodu odpowiedzialny na przeniesienie danych z logów do pliku .csv

4 Proces decyzyjny aktora

Aktor w pierwszej kolejności sprawdza ,czy nie znajduje się w stanie CANT_MOVE lub SAFE. W tym przypadku umieszcza on wpis w logach i automatycznie pomija swoją turę.

Następnie aktor weryfikuje położenie swoje względem ognia. Jeśli znajduje się on w ogniu to nadany jest mu status CANT_MOVE. Jeśli znajduje się on obok ognia to od jego aktualnego stanu zdrowia (wynoszącego domyślnie 10) odejmowane jest 3. Jeśli spowoduje to zejście statystyki zdrowia poniżej 0 to oznacza przejście w stan CANT_MOVE w innym przypadku wykonuje on ruch w losowe miejsce nie znajdujące się w ogniu lub będące obok ognia.

W kolejnym kroku sprawdzane jest czy aktor nie znajduje się w drzwiach. Jeśli się znajduje to w przypadku drzwi wyjścia z budynku jego status jest

ustawiany na SAFE ,a lokalizacja na (-10, -10), a w przypadku pozostałych typów drzwi po prostu podejmuje próbę przejścia do następnego pomieszczenia.

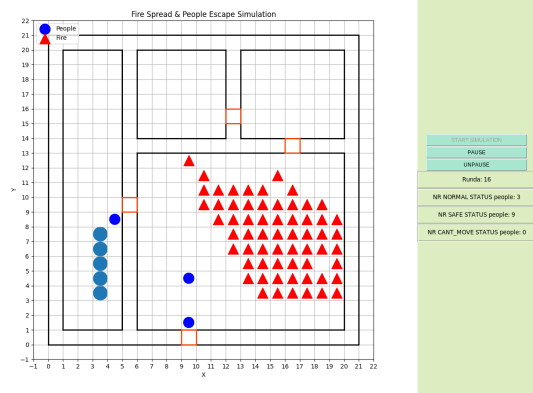
Jeśli w żadnym z powyższych kroków nie zakończy się tura aktora to podejmowana jest próba obrania strategii, reprezentowanej przez drzwi do których chce dotrzeć aktor. W tym procesie brane jest pod uwagę jakie drzwi widzi aktor, czy przez nie już przechodził i czy wie jakiego typu są to drzwi. Typy drzwi rozróżniane są na: DEADEND, TRANSITION, SIGNED_EXIT, UNSIGNED_EXIT. Poprzez test statystyki knowledge aktor może stwierdzić ,że dane drzwi prowadzą do pomieszczenia bez wyjścia lub do wyjścia z budynku.

W zależności od tego czy została obrana strategia to aktor będzie się albo poruszać zgodnie z nią, albo w względnie losowym kierunku.

5 Proces wizualizacji danych

Budynek został zwizualizowany w programie na podstawie pliku .yaml. Składa się on z 4 pokoi i 4 przejść, w czym jedno to wyjście z budynku. W budynku znajduje się pas przeszkód. Dane do symulacji zostały załadowane z pliku .csv i obrobione do postaci umożliwiającej symulację. Symulację zaczyna się kliknięciem na przycisk START SIMULATION. Można ją zapausować/odpauzować za pomocą przycisków PAUSE/UNPAUSE. Symulacja jest ustawiona na powtarzanie. Oprócz przycisków do sterowania animacją symulacji, mamy też dostęp do danych statystycznych o każdej rundzie: numer rundy

- ilość osób o statusie NORMAL
- ilość osób o statusie SAFE
- ilość osób o statusie CANT_MOVE



Rysunek 2: Wizualizacja przykładowego przebiegu symulacji

Rozprzestrzeniający się ogień jest wizualizowany za pomocą czerwonych trójkątów, zaś ludzie w zależności od statusu przyjmują postać kulek o różnych kolorach:

- status NORMAL - kolor niebieski
- status SAFE - kulka znika z pola planszy
- status CANT_MOVE - kolor czerwony
- status PANIC - kolor żółty